FOURIER Oktober 2013, Vol. 2, No. 2, 74 – 86

# METODE AKRA-BAZZI SEBAGAI GENERALISASI METODE MASTER DALAM MENYELESAIKAN RELASI REKURENSI

**M. Abrori**[1]

[1]Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

Jl. Marsda Adisucipto Yogyakarta, 55281

Email : borymuch@yahoo.com

***Abstract***

*Rekurensi relation is an equation that relates the elements of a sequence. One of the benefits of the rekurensi relation can be used to calculate the running time/finish of an algorithm. Some algorithms use approach devide-and-conquer in resolving a problem. Rekurensi relations with the approach of the devide and conquer can be solved by several methods. This research aims to know the Akra-Bazzi Method as an extension Method of the Master. This research began with the dissected the concept pertaining to the Relation Rekurensi, methods for resolving Relationship Rekurensi, and lastly about methods of Akra-Bazzi. Note that Akra-Bazzi Method can solve a rekurensi devide-and-conquer with shorter calculation.*

***Keywords:*** *relation rekurensi, devide-and-conquer, Master Method, method of Akra-Bazzi.*

## 1. PENDAHULUAN

Relasi rekurensi adalah sebuah persamaan yang menghubungkan elemen-elemen dari suatu barisan. Salah satu manfaat dari relasi rekurensi yaitu dapat digunakan dalam menyelesaikan/menghitung *running time* dari suatu algoritma. *Running time* adalah jumlah dari operasi atau banyaknya langkah operasi untuk n input.

Beberapa algoritma menggunakan pendekatan *devide-and-conquer* dalam menyelesaikan suatu permasalahan. Pendekatan divide-and-conquer memiliki tiga tahapan penyelesaian; membagi masalah kedalam beberapa sub-masalah yang mirip dengan masalah asli tetapi memiliki ukuran yang lebih kecil, menyelesaikan masing-masing sub-masalah secara rekursif, dan kemudian, jika diperlukan, menggabungkan penyelesaian dari masing-masing sub-masalah untuk menciptakan penyelesaian dari masalah asli. Dalam relasi rekurensi dapat berupa persamaan atau ketidaksamaan yang menjelaskan fungsi dalam bentuk input yang lebih kecil. Selain itu, relasi rekurensi juga dapat membagi sub-masalah dalam ukuran-ukuran yang sama atau pun tidak sama.

Relasi rekurensi dengan pendekatan devide-and-conquer dapat diselesaikan dengan beberapa metode. Adapun metode-metode dalam menyelesaikan relasi rekurensi tersebut diantaranya adalah:

- Metode Susbtitusi

Dalam metode substitusi, untuk menyelesaikan permasalahan rekurensi menggunakan dua langkah: menebak suatu bentuk penyelesaian (menggunakan pohon rekursi) dan menggunakan induksi matematika untuk menemukan konstanta dan menunjukkannya bahwa penyelesaian tersebut benar.

- Metode Pohon Rekursi

Pada metode pohon rekursi kita menggunakan/membuat pohon rekursi berupa garis-garis untuk merancang tebakan penyelesaian yang benar. Titik-titik pada pohon rekursi menggambarkan biaya pada tiap-tiap tempat sub-masalah tersebut.Dari masing-masing titik kitamenjumlahkan biaya pada tiap-tiap tingkat, dan kemudian menjumlahkan biaya semuatingkat untuk memperoleh total biaya pada semua tingkat rekursi.

- Metode Master

Metode *Master* merupakan metode yang sudah umum digunakan untuk menyelesaikan relasi rekurensi. Metode Master mudah digunakan apabila kita sudah berhasil menentukan jenis kasusnya. Akan tetapi metode ini terbatas penggunaannya. Metode Master tidak mampu untuk menyelesaikan relasi rekurensi dengan bentuk $T(n) = T(n-1) + \ldots$.

- Metode Akra-Bazzi

Metode Akra-Bazzi ditemukan oleh dua orang dari Beirut yang bernama Mohamad A Akra dan Louay M J Bazzi. Metode ini dipublikasikan dalam jurnal *Computational Optimization and Applications* pada bulan Mei 1998. Metode Akra-Bazzi lebih kuat atau lebih umum dari pada Metode Master. *The Master Method is fairly powerfull and result in a closed form solution for devide-and-conquer recurrences with a special (but commonly-occuring) form* [1].

Seperti yang sudah dinyatakan di atas bahwa metode master mudah digunakan apabila sudah bisa ditentukan jenis kasusnya. Metode ini juga terbatas penggunaannya karena tidak mampu menyelesaian relasi rekurensi dengan bentuk $T(n) = T(n-1) + \ldots$. *Recently Akra and Bazzi discovered a far more general solution to divide-and-conquer recurrences* [1]. Metode Akra-Bazzi lebih kuat dan umum dari pada Metode Master. Untuk itu,dengan penelitian ini

akan diketahui sejauh mana kelebihan Metode Akra-Bazzi dibandingkan dengan Metode Master.

Tujuan dari penelitian yang hendak dicapai yaituuntuk:

1. Menyelesaikan relasi rekurensi denganmetode Akra-Bazzi.
2. Mengetahui bentuk metode Akra-Bazzi sebagai generalisasi dari metode master dalam menyelesaikan relasi rekurensi.

Adapun kegunaan dari penelitian ini adalah untuk menghitung running time dari suatu algoritma berbasis devide-and-conquer. Dalam penelitian ini, akan diperlihatkan bagaimana keterbatasan metode master dibandingkan dengan metode Akra-Bazzi. Akhirnya akan diketahui bagaimana bentuk metode Akra-Bazzi sebagai generalisasi dari metode master.

## 2. METODE PENELITIAN

Suatu prosedur penyelesaian masalah untuk mencari kebenaran yang dituangkan dalam bentuk perumusan masalah, studi literatur, asumsi-asumsi dan hipotesis, pengumpulan dan penganalisaan data, hingga penarikan kesimpulan disebut dengan metodologi penelitian [2]. Dengan metodologi penelitian akan dihasilkan kajian materi penelitian yang lebih utuh dan komprehensif.

Penelitian ini merupakan studi literatur yang dilakukan dengan mempelajari beberapa karya ilmiah dalam bentuk jurnal maupun buku teks atau artikel-artikel lainnya yang menunjang penelitian tentang Metode Akra-Bazzi sebagai Generalisasi Metode Master dalam Menyelesaikan Relasi Rekurensi. Disamping itu, dengan memanfaatkan media *online*, peneliti juga melakukan pencarian (*searching*) literatur menggunakan Google Search (*search engine*) dengan memasukkan beberapa *keyword* yang mengacu pada relasi rekurensi, *master theorem*, *asymptotic notation*, algoritma, analisis algoritma dan metode Akra-Bazzi, dan kemudian mengunduh file (.doc, .pdf) yang sesuai/mendukung penelitian ini.

Penelitian ini dimulai dengan membedah konsep yang berkaitan dengan Relasi Rekurensi, metode-metode untuk menyelesaikan Relasi Rekurensi, dan terakhir tentang metode Akra-Bazzi.

## 3. TEORI

a. Kalkulus Integral

Integral adalah kebalikan dari proses diferensiasi. Integral ditemukan menyusul ditemukannya masalah dalam diferensiasi di mana matematikawan harus berpikir bagaimana

menyelesaikan masalah yang berkebalikan dengansolusi diferensiasi. Lambang integral adalah $\int$ . Integral suatu fungsi $f(x)$ secara matematis ditulis dan dinyatakan sebagai:

$$\int f(x)dx = F(x) + c \, .$$

Dalam hal ini:

- $f(x)$ disebut sebagai integran
- $F(x)$ disebut sebagai elemen integrasi
- $c$ disebut sebagai konstanta integrasi.

Integral terbagi dua yaitu integral tak tentu dan integral tertentu. Integral taktentu adalah integral yang mana nilai $x$ dari fungsi tidak disebutkan sehingga dapat menghasilkan nilai dari fungsi tersebut yang banyak, atau dengan kata lain tidak memiliki batas atas dan batas bawah. Sedangkan integral tertentu adalah integral yang mana nilai $x$ dari fungsi telah ditentukan, sehingga nilai dari fungsi integral tersebut terbatas pada nilai x yang telah ditetapkan tersebut (memiliki batas atas dan batas bawah).

Andaikan fungsi $f$ dan fungsi $g$ mempunyai anti turunan dan $k$ adalah suatu konstanta, maka (sifat kelinearan integral):

$$\int kf(x)dx = k\int f(x)dx$$

$$\int [f(x) \pm g(x)]dx = \int f(x)dx \pm \int g(x)dx$$

Andaikan fungsi $f$ kontinu (dapat diintegralkan) pada [a,b] dan $F$ sebarang anti turunan dari $f$, maka (teorema fundamental kalkulus):

$$\int_a^b f(x)dx = F(b) - F(a)$$

b. Analisis Algoritma

Dalam bidang matematika, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah secara logis dan sistematis. Algoritma adalah sesuatu yang eksplisit, tepat, jelas, urutan mekanis-eksekusi dari suatu instruksi [3]. Dalam menyelesaikan masalah, kumpulan perintah ini diterjemahkan dan diproses secara bertahap dari awal hingga akhir, dengan dipenuhinya kondisi awal sebelum menjalankan algoritma.

Analisis algoritma adalah suatu cabang khusus dalam ilmu komputer yang mempelajari karakteristik dan performa dari suatu algoritma dalam menyelesaikan masalah, terlepas dari implementasi algoritma tersebut. Dalam cabang disiplin ini algoritma dipelajari secara

abstrak, terlepas dari sistem komputer atau bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan kriteria yang sama [4].

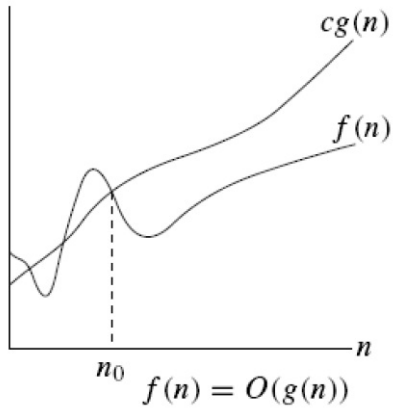
Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Dengan istilah lain, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sedangkan algoritma yang membutuhkan waktu lama untuk menyelesaikan masalahnya mempunyai kompleksitas yang tinggi. Suatu algoritma selain menyelesaikan suatu masalah juga harus mempertimbangkan mengenai faktor keefektifitan. Untuk mengetahui tentang keefektifan sebuah algoritma kita dapat menggunakan beberapa notasi berikut:

- Notasi $O$ besar

Notasi $O$ besar memberikan batas atas untuk fungsi ke dalam faktor konstan.

$$O(g(n)) = f(n)$$

untuk suatu bilangan positif $c$ dan $n_0$, maka $0 \le f(n) \le cg(n)$ untuk semua $n \ge n_0$. Dengan istilah lain, fungsi $f(n)$ akan selalu berada dibawah $cg(n)$ pada $n \ge n_0$.



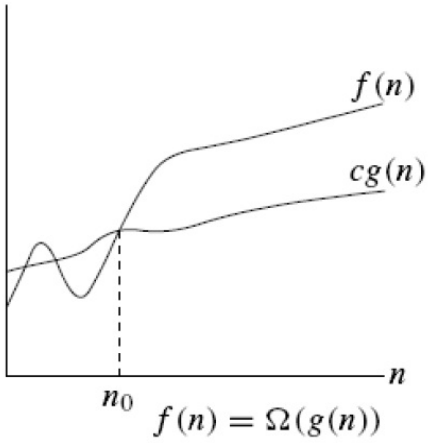

Gambar 1. Kurva $f(n)$ akan selalu berada dibawah $cg(n)$ pada $n \ge n_0$

- Notasi Omega besar $(\Omega)$

Seperti halnya pada notasi $O$ besar, Notasi $\Omega$ besar memberikan batas bawah untuk fungsi ke dalam faktor konstan.

$$\Omega(g(n)) = f(n)$$

untuk suatu bilangan positif $c$ dan $n_0$, maka $0 \le cg(n) \le f(n)$ untuk semua $n \ge n_0$.



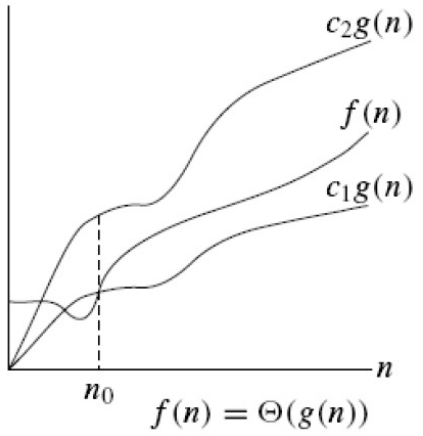

Gambar 2. Kurva $f(n)$ berada diatas $cg(n)$ pada $n \geq n_0$

- Notasi Tetha Besar $(\Theta)$

Pada notasi tetha besar $(\Theta)$ memberikan suatu fungsi suatu batas atas dan batas bawah. Secara matematis adalah sebagai berikut:

$$\Theta(g(n)) = f(n)$$

untuk suatu bilangan positif $c_1$, $c_2$ dan $n_0$, maka $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ untuk semua $n \geq n_0$.




Gambar 3. Kurva $f(n)$ berada diantara $c_1 g(n)$ dan $c_2 g(n)$ pada $n \geq n_0$.

c. Relasi Rekurensi

Disadari atau tanpa disadari permodelan yang menggunakan relasi rekurensi cukup banyak di sekitar kita. Sebagai contoh nyata seperti pada persoalan bunga deposito, populasi hewan, tabungan, dan lain-lain. Salah satu dari relasi rekurensi yang tertua adalah bilangan

Fibonacci, yang berawal dari suatu pertanyaan*:* "Sesudah satu tahun berapa banyak pasangan kelinci akan diperoleh apabila pada awal tahun terdapat sepasang kelinci, dan pada setiap bulan setiap pasangan melahirkan pasangan baru yang menjadi pasangan produktif sesudah satu bulan? Dengan asumsi bahwa tidak terjadi kematian dalam tahun tersebut".

Berikut ini adalah contoh sederhana dari relasi rekurensi: "Ikhram menabung uangnya di bank sebesar Rp. 100.000,- dengan bunga 12% per tahun. Bila $X_n$ adalah jumlah uang pada akhir tahun ke $n$, tentukan hubungan yang terdapat di antara $X_n$ dan $X_{n-1}$".

Pada akhir tahun ke $n-1$ jumlah uang Ikhram adalah $X_{n-1}$. Sesudah satu tahun, maka akan diperoleh jumlah $X_{n-1}$ ditambah dengan bunga tadi, sehingga,

$$X_n = X_{n-1} + (0{,}12)X_{n-1}$$

atau

$$X_n = 1{,}12X_{n-1} \ .$$

Ini disebut relasi rekurensi, dengan nilai awal $X_0 = 100.000$.

Berikut ini beberapa bentuk relasi rekurensi:

- Bilangan Fibonacci (0, 1, 1, 2, 3, 5, 8, 13, . . .)

  $F_n = F_{n-1} + F_{n-2}$ (disebut relasi rekurensi)

  untuk $n > 1$ dan kondisi awal $F_0 = 0$, $F_1 = 1$.
- $T(n) = 2T(n/2) + n$, Mergesort.
- $T(n) = 2T(n-1) + 1$, Menara Hanoi.

Adapun metode-metode untuk menyelesaikan relasi rekurensi diantaranya adalah:

- Metode Susbtitusi

Dalam Metode Substitusi, untuk menyelesaikan permasalahan rekurensi menggunakan dua langkah: menebak suatu bentuk penyelesaian (menggunakan pohon rekursi) dan menggunakan induksi matematika untuk menemukan konstanta dan menunjukkannya bahwa penyelesaian tersebut benar [5].

Contoh:

Akan dibuktikan bahwa penyelesaian rekurensi dari $T(n) = 2T(n/2) + n$ yaitu $T(n) = O(n \lg n)$.

Penyelesaian:

Dengan menggunakan Metode Substitusi akan ditunjukkan bahwa $T(n) \le cn\lg n$ untuk suatu konstanta $c > 0$.

Dimulai dengan mengasumsikan bahwa untuk semua bilangan positif $m < n$, khususnya untuk $m = \lfloor n/2 \rfloor$, memberi hasil

$$T(\lfloor n/2 \rfloor) \le c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor).$$

Hasil ini disubstitusikan ke persamaan rekurensi sehingga diperoleh:

$$\begin{aligned} T(n) &\le 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\ &\le cn\lg(n/2)` + n \\ &= cn\lg n - cn\lg 2 + n \\ &= cn\lg n - cn + n \\ &\le cn\lg n, \text{ dimana berlaku untuk } c \ge 0. \end{aligned}$$

Dengan menggunakan induksi matematika bisa ditunjukkan bahwa penyelesaian tersebut benar.

- Metode Pohon Rekursi

Pada metode pohon rekursi kita menggunakan/membuat pohon rekursi berupa garis-garis untuk merancang tebakan penyelesaian yang benar. Titik-titik pada pohon rekursi menggambarkan biaya pada tiap-tiap tempat sub-masalah tersebut. Dari masing-masing titik kita menjumlahkan biaya pada tiap-tiap tingkat, dan kemudian menjumlahkan biaya semua tingkat untuk memperoleh total biaya pada semua tingkat rekursi.

- Metode *Master*

Metode *Master* merupakan metode yang sudah umum digunakan untuk menyelesaikan relasi rekurensi. Metode *Master* bergantung pada teorema *Master*:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Dimana, $a \ge 1$, $b > 1$ konstan, dan $f(n)$ adalah fungsi non-negatif integer. Ada tiga kemungkinan:

(a). Jika $f(n) = O(n^{\log_b a - \varepsilon})$, untuk konstan $\varepsilon > 0$, maka $T(n) = \Theta(n^{\log_b a})$,

(b). Jika $f(n) = \Theta(n^{\log_b a})$, maka $T(n) = \Theta(n^{\log_b a} \lg n)$.

(c). Jika $f(n) = \Omega(n^{\log_b a + \varepsilon})$, untuk $\varepsilon > 0$, dan jika $af(n/b) \le cf(n)$, untuk konstan $c < 1$ dan semua nilai $n$ yang besar, maka $T(n) = \Theta(f(n))$.

Metode master mudah digunakan apabila kita sudah berhasil menentukan jenis kasusnya. Akan tetapi metode ini terbatas penggunaannya. Metode master tidak mampu untuk menyelesaikan relasi rekurensi dengan bentuk $T(n) = T(n-1) + \ldots$.

- Metode Akra-Bazzi

Metode Akra-Bazzi ditemukan oleh dua orang dari Beirut yang bernama Mohamad A. Akra dan Louay M. J. Bazzi. Metode ini dipublikasikan dalam jurnal *Computational Optimization and Applications* pada bulan Mei 1998. Metode Akra-Bazzi lebih kuat atau lebih umum dari pada Metode Master.

## 4. HASIL DAN PEMBAHASAN

Pada bagian 3 sudah dibedah konsep yang berkaitan dengan Relasi Rekurensi danmet ode-metode untuk menyelesaikan Relasi Rekurensi. Selanjutnya pada bagian D ini akan dikupas tentang konsep Metode Master dan Metode Akra-Bazzi beserta contoh soal dan penyelesainnya. Selanjutnya akan diketahui kelebihan Metode Akra-Bazzi dibandingkan dengan Metode Master

a. Metode Master

Thomas H. Cormen, dkk. [5] dalam bukunya yang berjudul Introduction to Algorithms ed. ke-3 tahun 2009 mengatakan bahwa metode master memberikan suatu metode "*cookbook*" untuk menyelesaikan rekurensi dari bentuk:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

dimana $a \geq 1$ dan $b > 1$ merupakan konstanta dan $f(n)$ merupakan fungsi asimtot positif. Untuk menggunakan metode master, permasalahan dibagi menjadi tiga kasus. Selanjutnya apabila sudah ditentukan kasusnya, maka dengan mudah rekurensi bisa diselesaikan. Metode master didasarkan pada Teorema Master.

Teorema 1

Misal $a \geq 1$ dan $b > 1$ merupakan suatu kontanta, $f(n)$ suatu fungsi, dan $T(n)$ didefinisikan pada bilangan bulat non negatif oleh rekurensi

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

dimana dianggap $\frac{n}{b}$ berarti merupakan salah satu dari $\left\lfloor \frac{n}{b} \right\rfloor$ atau $\left\lceil \frac{n}{b} \right\rceil$. Maka $T(n)$ mempunyai batas asimtotik sebagai berikut:

1. Jika $f(n) = O(n^{\log_b a - \varepsilon})$ untuk suatu konstanta $\varepsilon > 0$, maka $T(n) = \Theta(n^{\log_b a})$.
2. Jika $f(n) = \Theta(n^{\log_b a})$, maka $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Jika $f(n) = \Omega(n^{\log_b a + \varepsilon})$ untuk suatu konstanta $\varepsilon > 0$, dan jika $af(n/b) \le cf(n)$ untuk suatu konstanta $c < 1$ dan sembarang bilangan yang cukup besar $n$ n, maka $T(n) = \Theta(f(n))$. [5]

Penentuan satu dari tiga kasus di atas diperoleh dengan cara membandingkan fungsi $f(n)$ dengan fungsi $n^{\log_b a}$. Mana yang lebih besar itu merupakan penyelesaian dari rekurensi. Kasus pertama, jika fungsi $n^{\log_b a}$ lebih besar dari pada fungsi $f(n)$ maka penyesaiannya yaitu $f(n) = \Theta(n^{\log_b a})$. Kasus kedua, jika $n^{\log_b a}$ dan $f(n)$ sama ukurannya maka dikalikan dengan faktor logaritmis sehingga penyelesaiannya yaitu,

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(f(n) \lg n).$$

Kasus ketiga, jika fungsi $f(n)$ yang lebih besar maka penyelesaiannya yaitu $T(n) = \Theta(f(n))$.

Selain ketentuan-ketentuan di atas, diperlukan kecermatan terhadap hal-hal yang bersifat teknis. Dalam kasus pertama, tidak hanya $f(n)$ harus lebih kecil dari pada $n^{\log_b a}$, $f(n)$ juga harus merupakan polinomial yang lebih kecil. Dengan kata lain, $f(n)$ secara asimtot lebih kecil dari pada $n^{\log_b a}$ oleh faktor dari $n^{\varepsilon}$ untuk suatu konstanta $\varepsilon < 0$. Di dalam kasus ketiga, tidak hanya $f(n)$ harus lebih besar dari pada $n^{\log_b a}$, tetapi juga harus secara polinomial lebih besar dan dalam pertambahan memenuhi kondisi bahwa $af(n/b) \le cf(n)$.

Ada tiga kasus yang tidak memenuhi kemungkinan-kemungkinan untuk $f(n)$. Terdapat gap antara kasus 1 dan 2 ketika $f(n)$ lebih kecil dari pada $n^{\log_b a}$ tetapi bukan merupakan polinomial yang lebih kecil. Serupa dengan hal tersebut, terdapat gap antara kasus 2 dan 3 ketika $f(n)$ lebih besar dari pada $n^{\log_b a}$ tetapi secara polinomial tidak lebih besar. Jika fungsi $f(n)$ termasuk ke dalam salah satu dari gap tersebut, atau jika tidak memenuhi kondisi yang dipersyaratkan untuk kasus 3, maka metode master tidak dapat digunakan untuk menyelesaikan rekurensi.

Di bawah ini diberikan contoh rekurensi yang diselesaikan dengan metode master:

Selesaikan rekurensi $T(n) = 16T\left(\frac{n}{4}\right) + n$.

Penyelesaian:

$a = 16$, $b = 4$, $f(n) = n$

$n^{\log_b a} = n^{\log_4 16} = \Theta(n^2)$

Karena $f(n) = O(n^{\log_4 16 - \varepsilon})$, dimana $\varepsilon = 1$, sehingga metode master bisa diterapkan dan menghasilkan penyelesaian $T(n) = \Theta\left(n^2\right)$.

b. Metode Akra-Bazzi

Metode Akra-Bazzi pertama kali dipublikasikan oleh peneliti yang berasal dari Libanon yaitu Mohamad Akra dan Louay Bazzi pada tahun 1998. Misalkan bentuk umum relasi rekurensi devide-and-conquer sebagai berikut:

$$T(n) = \sum_{i=1}^{k} a_i T\left(\frac{n}{b_i}\right) + f(n),$$

dimana $k$ merupakan suatu konstanta, $a_i > 0$ dan $b_i > 0$ konstan untuk semua nilai $i$, dan $f(n) = \Omega(n^c)$ dan $f(n) = O(n^c)$ untuk suatu konstanta $0 < c \le d$. Dalam hal ini diasumsikan bahwa $T(\theta(1)) = \theta(1)$. Akra dan Bazzi membuktikan bahwa rekurensi ini mempunyai penyelesaian asimtotik bentuk tertutup sebagai berikut:

$$T(n) = \theta\left( n^{\rho}\left(1 + \int_1^n \frac{f(u)}{u^{\rho+1}} du\right)\right),$$

dimana $\rho$ merupakan penyelesaian riil yang tunggal dari persamaan:

$\sum_{i=1}^{k} a_i / b_i^{\rho}$.

Implikasi dari Teorema Akra-Bazzi mengikuti bentuk dari Teorema Master, yakni:

$$T(n) = aT\left(\frac{n}{b}\right) + n^c \Rightarrow T(n) = \begin{cases} \theta(n^{\log_b a}) & \text{jika } c < \log_b a - \varepsilon \\ \theta(n^c \log n) & \text{jika } c = \log_b a \\ \theta(n^c) & \text{jika } c > \log_b a + \varepsilon \end{cases}$$

Teorema Akra-Bazzi tidak memerlukan parameter $a_i$ dan $b_i$ bernilai integer atau bilangan rasional genap. Di sisi lain, seandainyapun semua parameternya integer, persamaan karakteristik $\sum_{i=1}^{k} a_i / b_i^{\rho} = 1$ tidak mempunyai penyelesaian secara analitik.

Di bawah ini diberikan dua contoh rekurensi yang sulit apabila diselesaikan dengan cara pohon rekursi tetapi menjadi mudah penyelesaiannya dengan menggunakan Metoda Akra-Bazzi:

1. Quicksort Random: $T(n) = T(3n/4) + T(n/4) + n$.

   Persamaan $(3/4)^\rho + (1/4)^\rho = 1$ mempunyai penyelesaian tunggal $\rho = 1$, sehingga

   $$T(n) = \theta\left( n\left( 1 + \int_1^n \frac{1}{u} du \right) \right) = O(n \log n).$$

2. Seleksi deterministik: $T(n) = T(n/5) + T(7n/10) + n$

   Persamaan $(1/5)^\rho + (7/10)^\rho = 1$ tidak mempunyai penyelesaian secara analitik. Akan tetapi, bisa dilihat bahwa $(1/5)^x + (7/10)^x = 1$ merupakan fungsi menurun dari $x$, oleh karena itu $0 < \rho < 1$. Selanjutnya, diperoleh:

   $$\int_1^n \frac{f(u)}{u^{\rho+1}} du = \int_1^n u^{-\rho} du = \frac{u^{1-\rho}}{1-\rho}\Big|_{u=1}^{n} = \frac{u^{1-\rho} - 1}{1-\rho} = \theta(n^{1-\rho}),$$

   sehingga

   $T(n) = \theta(n^\rho . (1 + \theta(n^{1-\rho}))) = \theta(n)$. [6]

Dari contoh-contoh perhitungan di atas terlihat bahwa metode Akra-Bazzi dapat menyelesaikan suatu rekurensi devide-and-conquer dengan perhitungan yang singkat. Metode ini juga bisa menyelesaikan relasi rekurensi dengan bentuk $T(n) = T(n-1) + ...$ dimana kasus ini tidak bisa diselesaikan dengan Metode Master.

## 5. KESIMPULAN

Berdasarkan hasil pembahasan di atas diperoleh kesimpulan sebagai berikut:

1. Penyelesaian rekurensi dengan Metode Akra-Bazzi memberikan pengerjaan yang lebih mudah dan lebih singkat dibandingkan dengan Metode Master.
2. Bentuk umum Metode Akra-Bazzi sebagai generalisasi dari Metode Master dalam menyelesaikan relasi rekurensi, yaitu:

   $$T(n) = \theta\left( n^\rho \left( 1 + \int_1^n \frac{f(u)}{u^{\rho+1}} du \right) \right)$$

   dimana $\rho$ merupakan penyelesaian riil yang tunggal dari persamaan,

   $\sum_{i=1}^{k} a_i / b_i^\rho$.

## 6. DAFTAR PUSTAKA

[1] Leighton, Tom. 1996. *Notes on Better Master Theorems for Divide-and-Conquer Recurrences*. A journal. Cambridge: The MIT Press

[2] Subana, M. dan Sudrajat. 2001. *Dasar-Dasar Penelitian Ilmiah.* Bandung: CV. Pustaka Pelajar.

[3] www.cs.uiuc.edu/%7Ejeffe/teaching/algorithms/notes/00-intro.pdf tanggal akses 26 July 2012 Jam 13:05 WIB.

[4] http://id.wikipedia.org/wiki/Algoritma tanggal akses 25 July 2012 Jam 20:35 WIB.

[5] Cormen, Thomas H., Leiserson,Charles E, Rivest, Ronald L., Stein, Clifford, 2009. *Introduction to Algorithms*, Third Edition. London: The MIT Press.

[6] Erikson, Jeff, 2010. *Solving Recurrences*. http://www.cs.uiuc.edu/~jeffe/teaching /algorithms/